Question 1:

Number game between user and computer. The user starts by entering either 1 or 2 or 3 digits starting from 1 sequentially. The computer can return either 1 or 2 or 3 next digits in sequence, starting from the max number played by the user. User enters the next 1 or 2 or 3 next digits in sequence, starting from the max number played by the computer. Whoever reaches 20 first wins the game.

Note:

- the numbers should be in sequence starting from 1.

- minimum number user or computer should pick is at least 1 digit in sequence

- maximum number user or computer can pick only 3 digits in sequence

Example 1:

Player: 1 2

Computer played: [3, 4]

Player: 5 6 7

Computer played: [8, 9]

Player: 10

Computer played: [11, 12, 13]

Player: 14 15

Computer played: [16, 17, 18]

Player: 19 20

Player Wins!!!

Example 2:

Player: 1

Computer played: [2, 3]

Player: 4 5

Computer played: [6, 7, 8]

Player: 9 10

Computer played: [11]

Player: 12

Computer played: [13]

Player: 14 15

Computer played: [16]

Player: 17 18

Computer played: [19, 20]

Computer Wins!!!

Solution:

```
import random
def computer_turn(current_number):
    next_count = random.randint(1, 3)
    next_numbers = list(range(current_number + 1, current_number + next_count + 1))
    print(f"Computer played: {next_numbers}")
    return next_numbers
def user_turn(current_number):
    while True:
        try:
         user_input = input("Player: ")
            user_numbers = list(map(int, user_input.split()))
            if len(user_numbers) < 1 or len(user_numbers) > 3:
                print("You must enter 1, 2, or 3 numbers in sequence.")
                continue
            if user_numbers[0] != current_number + 1 or not all(
                    user_numbers[i] == user_numbers[i - 1] + 1 for i in range(1, len(user_numbers))):
                print("The numbers must be sequential and start from the last played number + 1.")
                continue
            return user_numbers
        except ValueError:
            print("Invalid input. Please enter numbers separated by spaces.")
```

```
def play_game():
    current_number = 0
    while True:
        user_numbers = user_turn(current_number)
        current_number = user_numbers[-1]
        if current_number >= 20:
            print("Player Wins!!!")
            break
        computer_numbers = computer_turn(current_number)
        current_number = computer_numbers[-1]
        if current_number >= 20:
            print("Computer Wins!!!")
            break
print("Welcome to the Number Game! The goal is to reach 20 first.")
print("Each turn, you can enter 1, 2, or 3 sequential numbers starting from the last number played.")
play_game()
```

Question 2:

Develop a function called ncr(n,r) which computes r-combinations of n-distinct object . use this function to print pascal triangle, where number of rows is the input

The formula for combinations is:

$$nCr = \frac{n!}{r!(n-r)!}$$

Solution:

```
def factorial(num):
    if num == 0 or num == 1:
        return 1
    result = 1
    for i in range(2, num + 1):
```

```
        result *= i
    return result

def ncr(n, r):
    return factorial(n) // (factorial(r) * factorial(n - r))

def triangle(rows):
    for n in range(rows):
        print(" " * (rows - n), end=" ")
        for r in range(n + 1):
            print(ncr(n, r), end=" ")
        print() # New line after each row
rows = int(input("Enter the number of rows for Pascal's Triangle: "))
triangle(rows)
```

Question 3:

Read a list of n numbers during runtime. Write a Python program to print the repeated elements with frequency count in a list.

Example :

Input:- [2,1,2,3,4,5,1,3,6,2,3,4]

Output:-

Element 2 has come 3 times

Element 1 has come 2 times

Element 3 has come 2 times

Element 4 has come 2 times

Element 1 has come 1 times

Element 6 has come 1 times

Solution:

```
from collections import Counter

numbers = list(map(int, input("Enter numbers separated by spaces: ").split()))

frequency = Counter(numbers)

for element, count in frequency.items():
```

```
    print(f"Element {element} has come {count} times")
```

Question 4:-

Develop a python code to read matric A of order 2X2 and Matrix B of order 2X2 from a file and perform the addition of Matrices A & B and Print the results.

File name matrices.txt

 # Matrix A

1 2

3 4

 # Matrix B

5 6

7 8

File main.py

Solution:

```
def read_matrices_from_file(filename):
    with open(filename, 'r') as file:
        lines = file.readlines()
        A = [list(map(int, lines[0].split())), list(map(int, lines[1].split()))]
        B = [list(map(int, lines[2].split())), list(map(int, lines[3].split()))]
    return A, B
def add_matrices(A, B):
    result = [[A[i][j] + B[i][j] for j in range(2)] for i in range(2)]
    return result
def print_matrix(matrix):
    for row in matrix:
        print(" ".join(map(str, row)))
A, B = read_matrices_from_file('matrix_input.txt')
```

```
result_matrix = add_matrices(A, B)
print("Matrix A:")
print_matrix(A)
print("\nMatrix B:")
print_matrix(B)
print("\nSum of A and B:")
print_matrix(result_matrix)
```

Question 5:-

Write a program that overloads the + operator so that it can add two objects of the class Fraction. Fraction can be considered of the for P/Q where P is the numerator and Q is the denominator

Solution:

```
import math
class Fraction:
    def __init__(self, numerator, denominator):
        self.numerator = numerator
        self.denominator = denominator
        self.simplify()

    def simplify(self):
        gcd = math.gcd(self.numerator, self.denominator)
        self.numerator //= gcd
        self.denominator //= gcd

    def __add__(self, other):
        if isinstance(other, Fraction):
            # Calculate the numerator and denominator for the result
```

```
            new_numerator = self.numerator * other.denominator + other.numerator * self.denominator
            new_denominator = self.denominator * other.denominator
            return Fraction(new_numerator, new_denominator)
        else:
            raise TypeError("Can only add two Fraction objects")
    def __str__(self):
        return f"{self.numerator}/{self.denominator}"
fraction1 = Fraction(1, 2)
fraction2 = Fraction(1, 3)
result = fraction1 + fraction2
print("Result of addition:", result)
```